

HTML5/CSS3

RESPONSIVE WEB ET PREPROCESSEUR SASS

Perspective et transformations en CSS

PROGRAMME:

1/ Rappel des transformatons en 2D:

1.1/Point d'origine:

1.2/Rotation:

1.3/Translation:

1.4/Scale:

2/ La perspective 3D:

2.1/ Percpective:

2.1/ Transformations:

3/ Tutorielle: Animer un cube:

1/ Rappel des transformations en 2D:

Nous avons vu que les transformations permettent de modifier l'emplacement, la rotation et la taille, selon les deux axes X et Y.

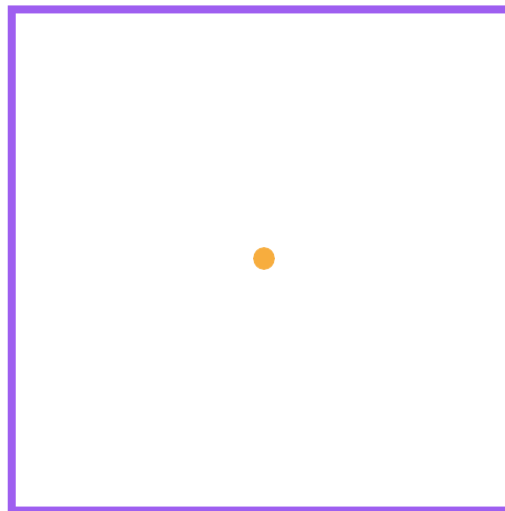


- 1- Sur l'axe X, les valeurs augmentent de gauche à droite.
- 2 Sur l'axe Y, les valeurs augmentent de haut en bas.

1/ Rappel des transformations en 2D:

1.1/ Point d'origine:

Les transformations d'objets sont appliquées par rapport à son point d'origine. Ce point est par défaut situé au centre l'objet.

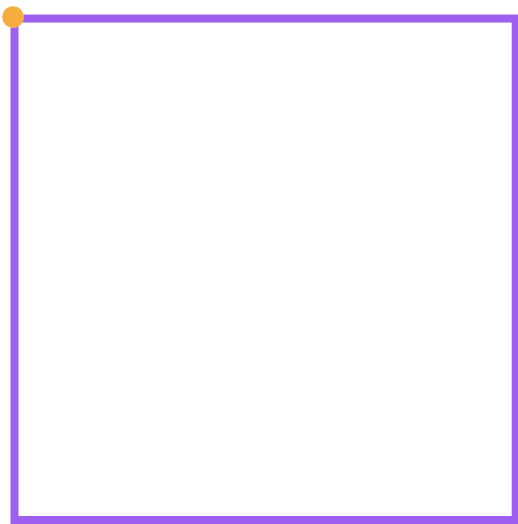


La propriété transform-origin permet de modifier le point d'origine d'un objet HTML. Elle accepte comme valeur le X et le Y de l'origine souhaité séparés par un espace. Ces deux informations peuvent être spécifiées en pourcentage, en pixel, ou par les mots clé left, right, top, bottom et center.

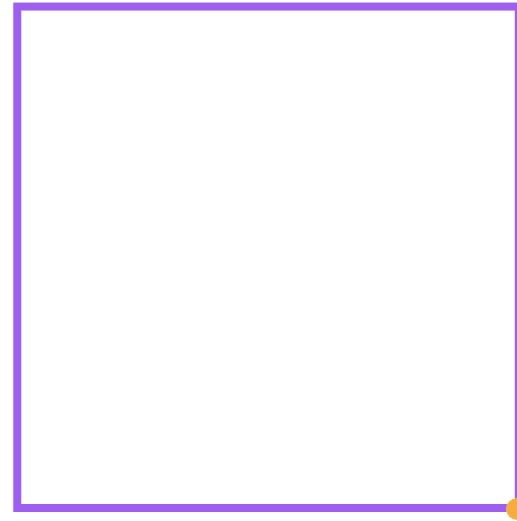
1/ Rappel des transformations en 2D:

1.1/ Point d'origine:

```
transform-origin:0 0;
```



```
transform-origin:100% 100%;
```



1/ Rappel des transformations en 2D:

Prenons pour exemple un simple carré, afin d'illustrer ce qu'il se passe:

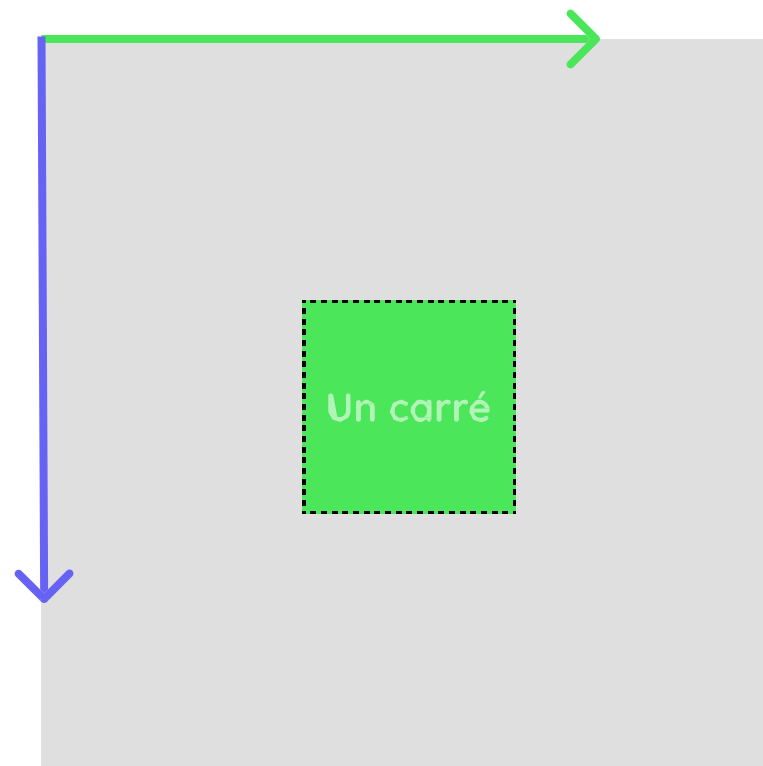
```
<div class="square">
  <h2>
    Un carrée
  </h2>
</div>
```

```
.square {
  background-color: ■ rgb(61, 194, 27);
  width: $square-size;
  height: $square-size;
}
```

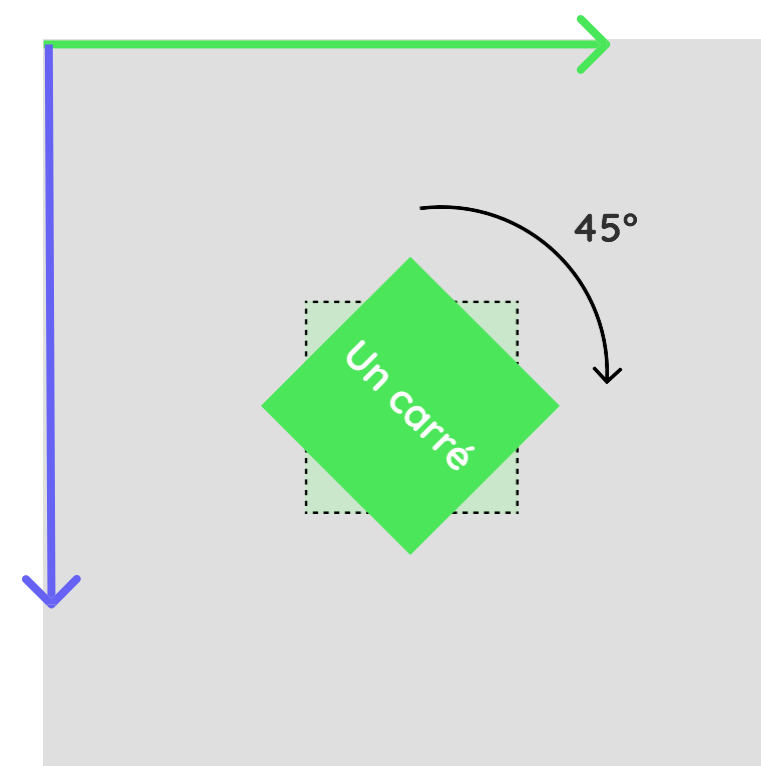
Il existe 3 sortes de transformations:

1.2/ La Rotation:

Permet de modifier la l'angle de rotation du carré.



transform:none (default)

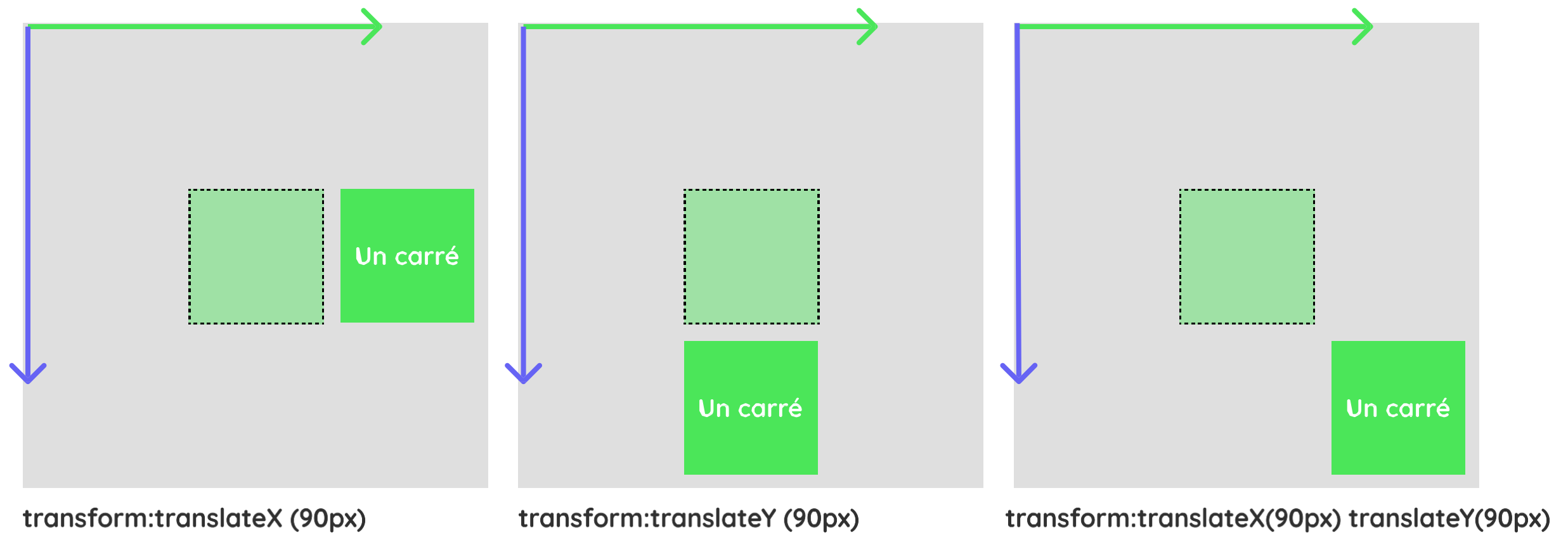


transform:rotate (45deg)

1/ Rappel des transformations en 2D:

1.3/ La translation: :

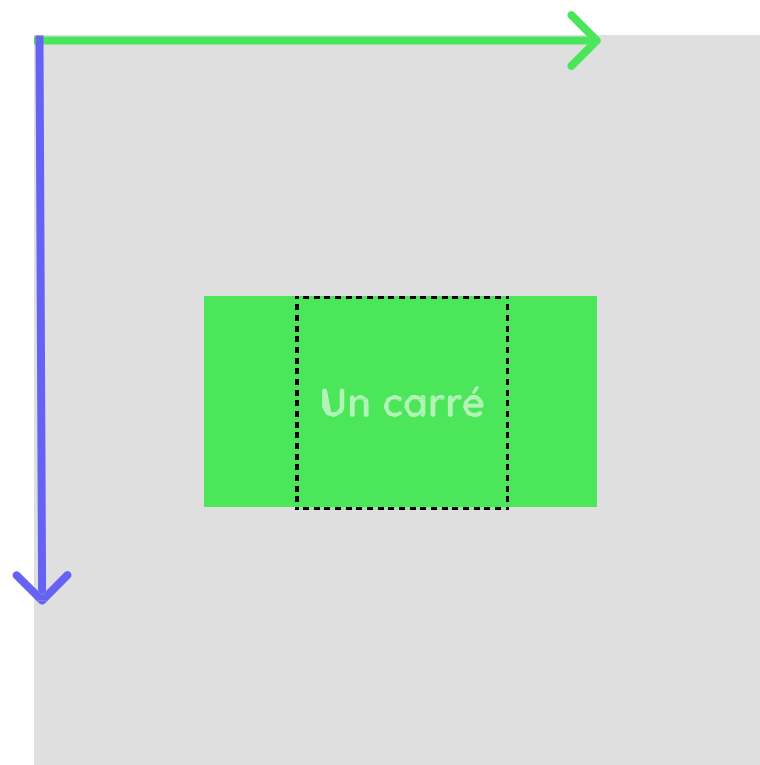
Permet de modifier la position du carré.



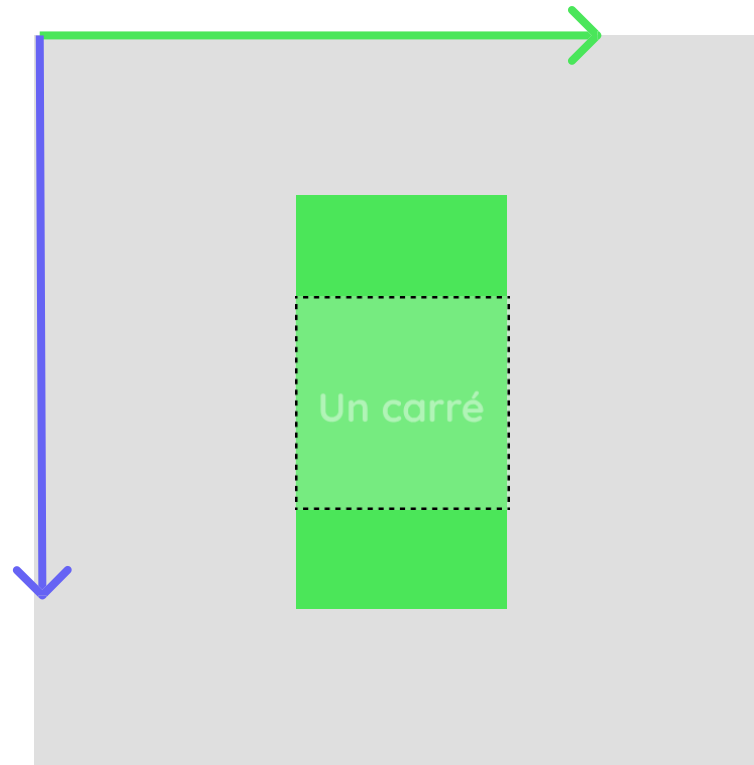
1/ Rappel des transformations en 2D:

1.4/ Scale :

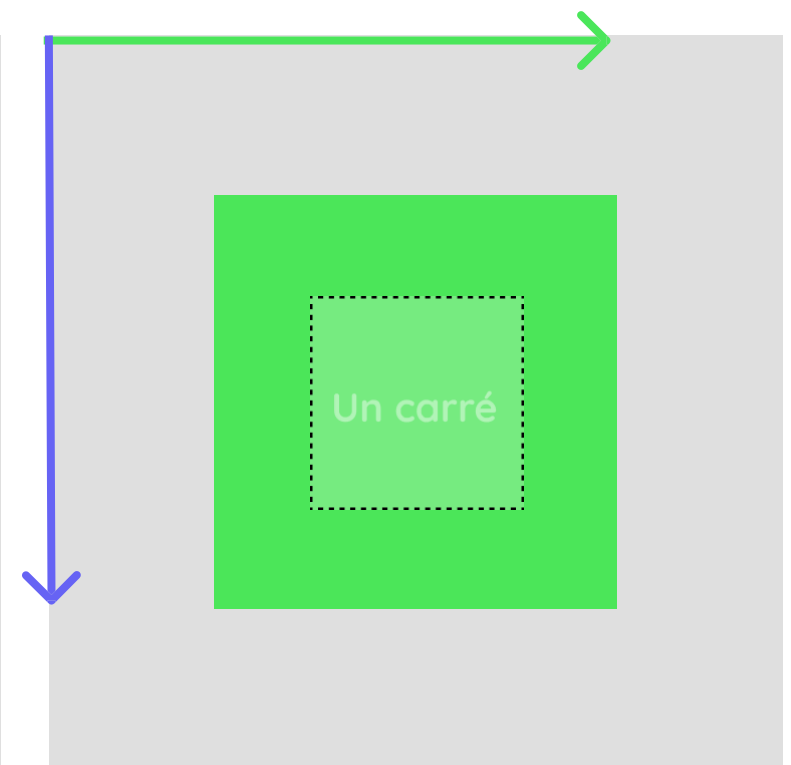
Permet de modifier la taille du carré.



`transform:scaleX (90px)`



`transform:scaleY (90px)`



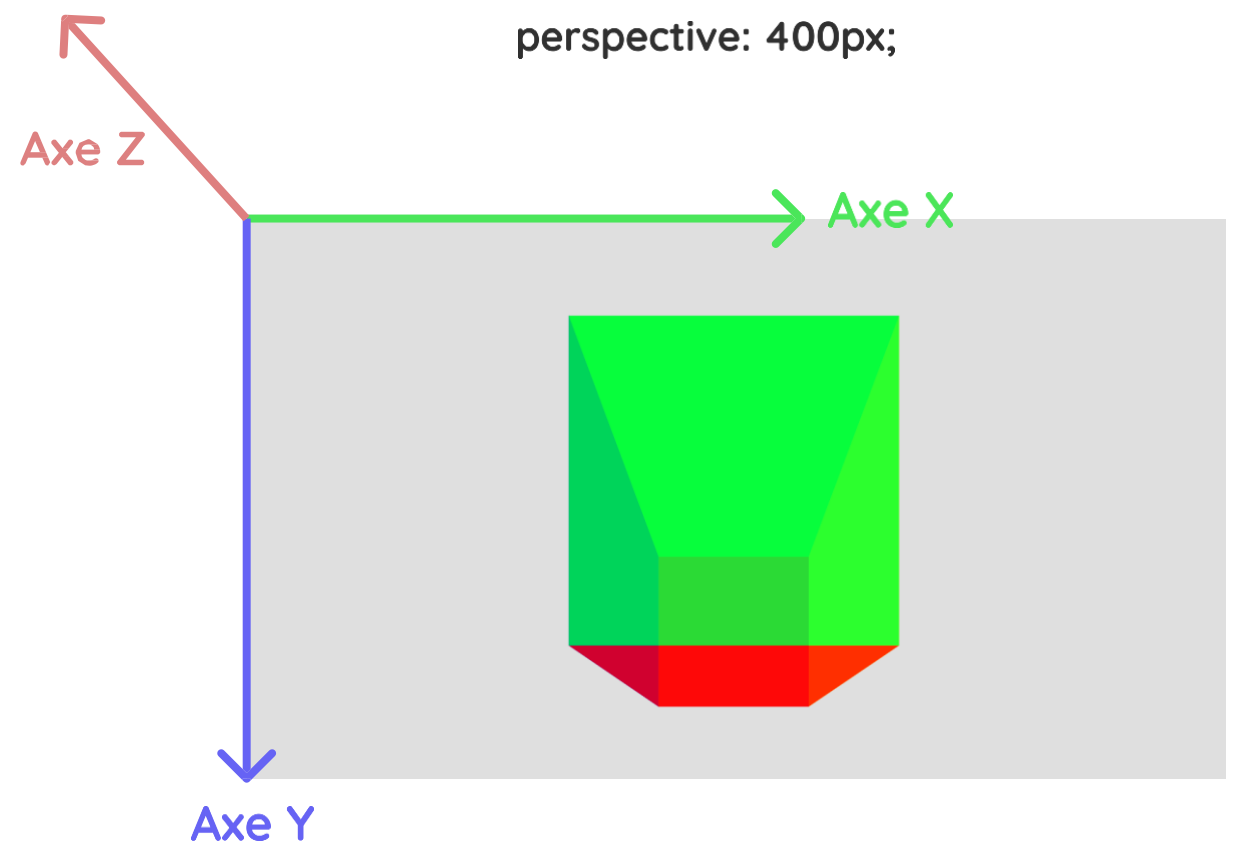
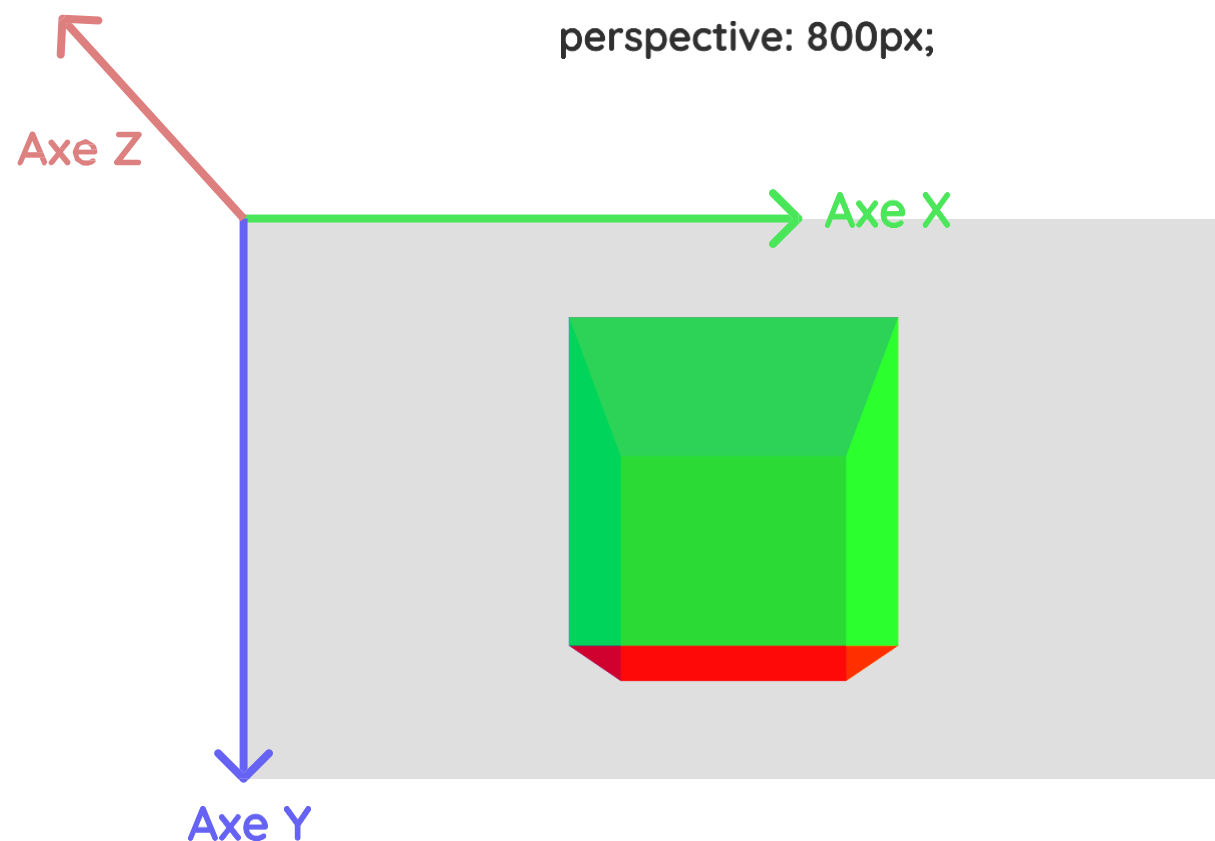
`transform:scaleX(90px) scaleY(90px)`

Transformation et animation en CSS

1/ La perspective 3D:

2.1/ Perspective:

La propriété CSS perspective permet d'activer la perspective 3D, en spécifiant la distance entre l'observateur/camera, et le point d'origine $z=0$, sur l'axe de perspective Z.



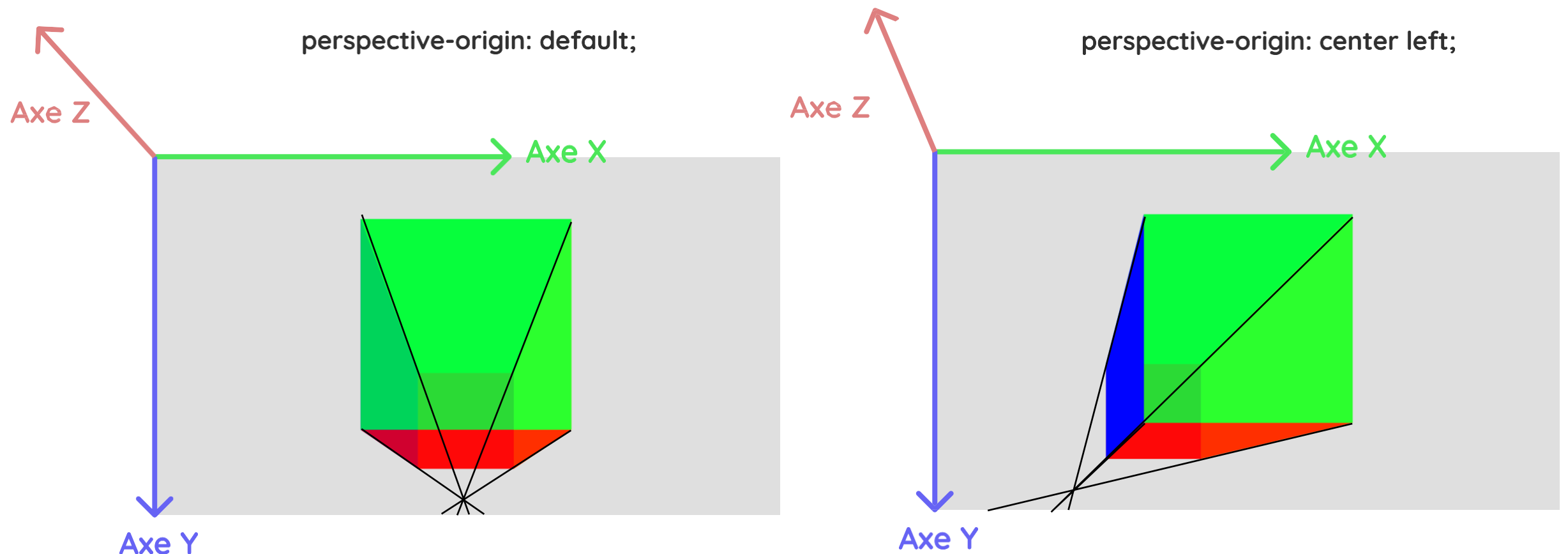
- Plus la valeur de la perspective est petite, plus l'effet perspective 3D est accentué.
- Sur l'axe Z, plus l'on s'approche de l'observateur/camera, plus les valeurs augmentent.

Transformation et animation en CSS

2/ La perspective 3D:

2.2/ Poit de fuite:

La propriété CSS `perspective-origin`, permet de spécifier la position du point de fuite sur l'axe X et Y dans le contexte 3D. Par default la valeur est au centre (50% 50%).



2/ La perspective 3D:

2.3/ Transformation 3D:

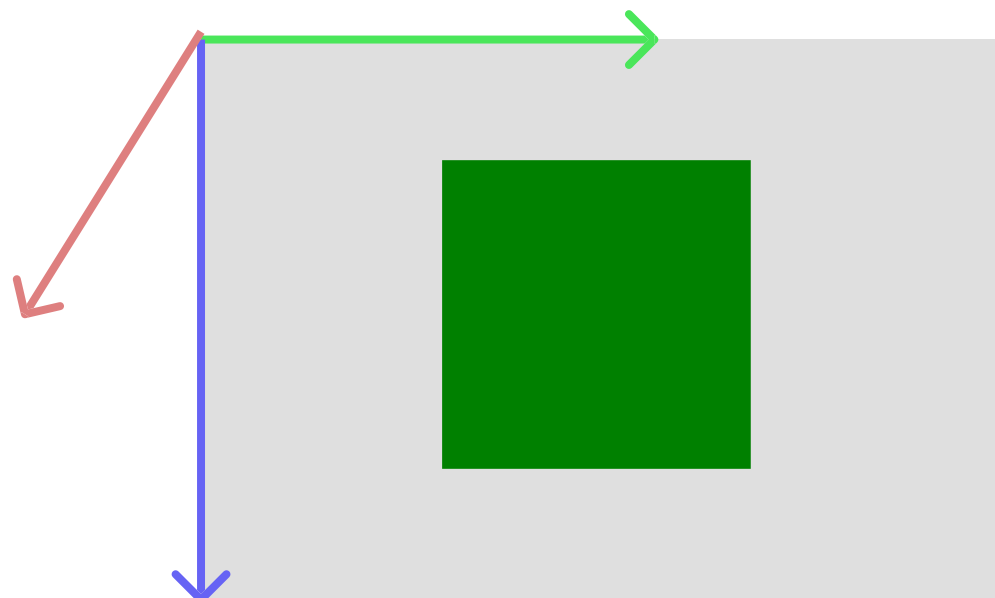
Reprenons l'exemple du carré, mais cette fois, nous déterminerons une perspective au parent, afin de pouvoir transformer le carré sur les 3 dimensions.

```
<div class="transformations">
  <div class="square">
    <h2>
      Un carré
    </h2>
  </div>
</div>
```

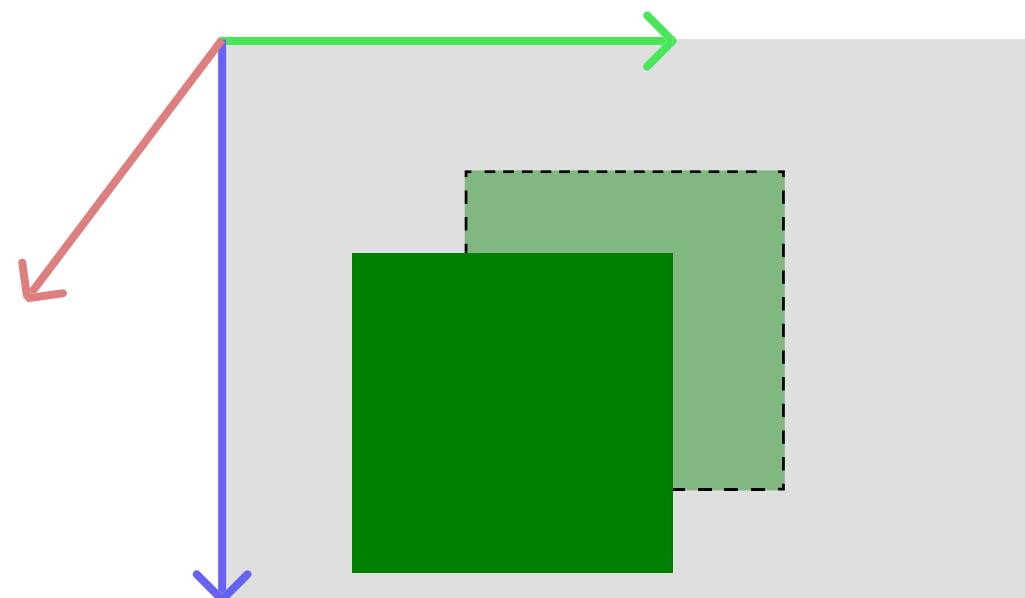
```
$square-size: 300px;
.transformations {
  perspective: 400px;
}
.square {
  background-color: green;
  width: $square-size;
  height: $square-size;
}
```

La translation du l'axe Z:

Cela permet de modifier la position sur l'axe Z. Si la position est supérieure à la profondeur déterminée avec la perspective, l'objet se retrouvera derrière l'observateur, et ne sera plus visible.



transform:none (default)



transform:translate3d (20px, 20px, 20px)

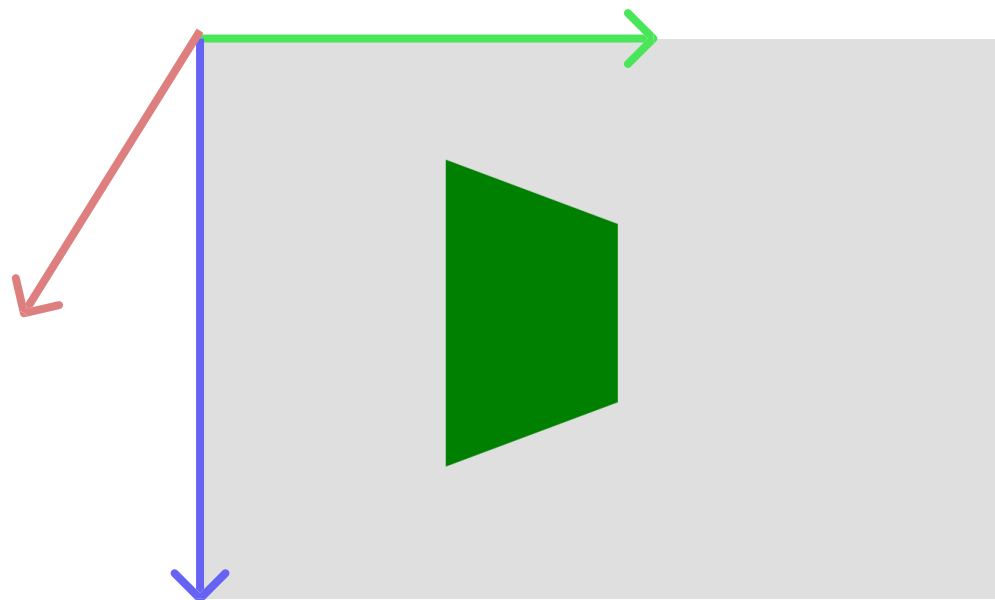
2/ La perspective 3D:

2.3/ Transformation 3D:

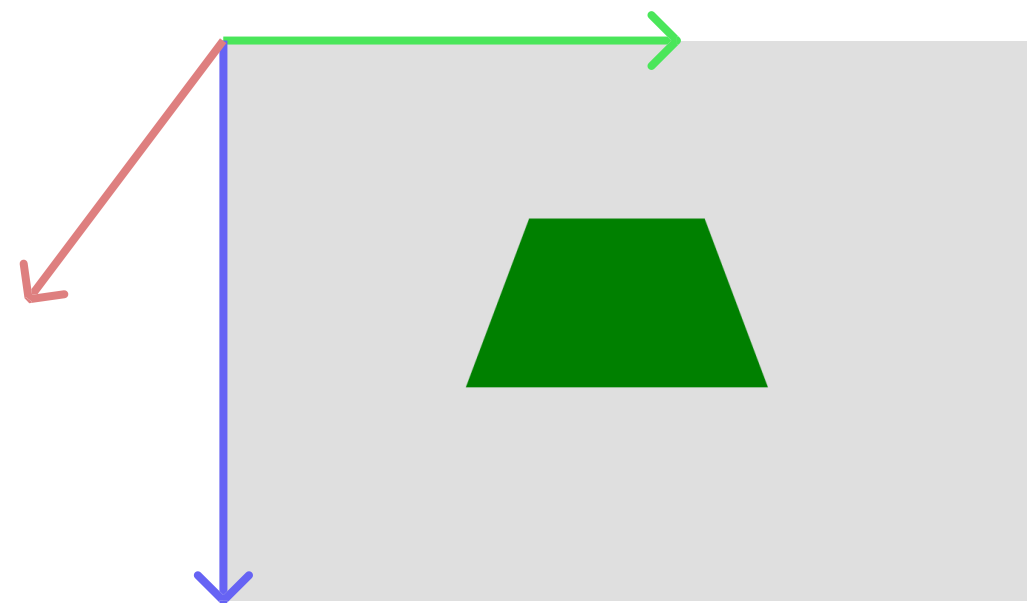
La Rotation:

La propriété `rotate()` et `rotateZ()` effectuent la même transformation.

Maintenant, nous pouvons aussi modifier l'angle correspondant les axes X et Y:



`transform: rotateY (45deg)`



`transform: rotateX (45deg)`

3/ Tutorielle: Animer un cube

Afin de reproduire un cube, nous allons représenter chaque face du cube par un <div>.

```
<!-- cube 3D -->
<div class="wrap">
  <div class="cube">
    <div class="front">
      <h2>1 perspective</h2>
    </div>
    <div class="right">
      <h2>2 transformation</h2>
    </div>
    <div class="back">
      <h2>3 transition</h2>
    </div>
    <div class="left">
      <h2>4 animation</h2>
    </div>
  </div>
</div>
```

```
.wrap{
  perspective: 800px;
  perspective-origin: 50% 25%;
  min-height: 100vh;
  display: flex;
  color: white;
}

.cube{
  margin: auto;
  position: relative;
  width: 300px;
  height: 300px;
  transform-style: preserve-3d;
  transform-origin: center;
  animation: cube-spin 5s ease-in-out infinite;
}

.cube div{
  position: absolute;
  background-color: rgba(1,0,12,0.726);
  width: 300px;
  height: 300px;
}
```

3/ Tutorielle: Animer un cube

Puis grâce aux transformations nous allons placer les quatre faces de manière à former un cube.

```
✓ .front{  
  transform: translateZ(150px);  
}  
  
✓ .right{  
  transform: rotateY(90deg) translateX(150px);  
  transform-origin: top right;  
}  
  
✓ .left{  
  transform: rotateY(-90deg) translateX(-150px);  
  transform-origin: center left;  
}  
  
✓ .back{  
  transform: translateZ(-150px) rotateY(180deg);  
}
```



3/ Tutorielle: Animer un cube

Nous allons ajouter une animation, qui fera tourner le cube sur l'axe Y.

```
@keyframes cube-spin {  
  0% {  
    transform: rotate3d(0, 0, 0, 0);  
  }  
  20% {  
    transform: rotate3d(0, 1, 0, 90deg);  
  }  
  30% {  
    transform: rotate3d(0, 1, 0, 90deg);  
  }  
  50% {  
    transform: rotate3d(0, 1, 0, 180deg);  
  }  
  60% {  
    transform: rotate3d(0, 1, 0, 180deg);  
  }  
  80% {  
    transform: rotate3d(0, 1, 0, 270deg);  
  }  
  100% {  
    transform: rotate3d(0, 1, 0, 360deg);  
  }  
}
```